=> d his

(FILE 'USPAT' ENTERED AT 16:00:27 ON 24 AUG 95)
L1 828 S 395/700/CCLS
L2 1426 S 364/280,281.6/CCLS
L3 1972 S L1 OR L2
L4 1281 S ACCESS AND SERVER AND (USER OR CLIENT)
L5 336 S ACCESS(P)SERVER(P)(USER OR CLIENT)
L6 39 S RESOURCE(P)L5
L7 669 S RESOURCE(10A)(INFORMATION OR DESCRIPTION OR STUB)
L8 102 S L4 AND L7

=> d ti,ab 49,52,53

US PAT NO: 5,271,007 [IMAGE AVAILABLE] L8: 49 of 102
TITLE: Network system having controlled access to available resources

ABSTRACT:
A network system in which a plurality of information devices connected to each other through a network are provided so that a resource is released to the information devices through the network, the system comprising: a storage device for storing an access list indicating an access right for every information device of release destination; and a management device for releasing a resource within a range in accordance with a right given to every information device on the basis of the access list stored in the storage device.

US PAT NO: 5,263,158 [IMAGE AVAILABLE] L8: 52 of 102
TITLE: Method and system for variable authority level user access control in a distributed data processing system having multiple resource manager

ABSTRACT:
Variable authority level user access control for a plurality of resource objects within a distributed data processing system having a plurality of resource managers. A reference monitor service is established and a plurality of access control profiles are stored therein, each including an identification of a selected user and a specified level of authority associated with that selected user . Thereafter, selected access control profiles are exchanged between the reference monitor service and a resource manager in response to an attempted access of a particular resource object controlled by that resource manager. The resource manager may then control access to the resource object by utilizing the exchanged access control profile to determine the extent access is permitted by means of the specified level of authority contained therein. In a preferred embodiment of the present invention, the access intent of a selected user is determined in conjunction with an attempted access of a particular resource object and stored. Thereafter, a comparison of the stated access intent with the specified level of authority contained within the access control profile may be utilized to grant or deny access .

US PAT NO: 5,263,157 [IMAGE AVAILABLE] L8: 53 of 102
TITLE: Method and system for providing user access control within a distributed data processing system by the exchange of access control profiles

ABSTRACT:

A method is disclosed for providing user access control for a plurality of resource objects within a distributed data processing system having a plurality of resource managers. A reference monitor service is established and a plurality of access control profiles are stored therein. Thereafter, selected access control profiles are exchanged between the reference monitor service and a resource manager in response to an attempted access of a particular resource object controlled by that resource manager. The resource manager may then control access to the resource object by utilizing the exchanged access control profile. In a preferred embodiment of the present invention, each access control profile may include access control information relating to a selected user ; a selected resource object; a selected group of users; a selected set of resource objects; or, a predetermined set of resource objects and a selected group of users.

=> b
=> d ab

US PAT NO: 5,263,165 [IMAGE AVAILABLE] L10: 1 of 1

ABSTRACT:
The method of the present invention may be utilized to provide user access control for a plurality of resource objects within a distributed data processing system having a plurality of resource managers. A reference monitor service is established and a plurality of access control profiles are stored therein. Thereafter, selected access control profile information may be communicated between the reference monitor service and a resource manager in response to an attempted access of a particular resource object controlled by that resource manager. A resource manager may utilize this communication technique to retrieve, modify, or delete a selected access control profile , as desired. Further, the resource manager may utilize this communication technique to control access to a resource object by utilizing the information contained within the access control profile to determine if the requester is authorized to access the resource object and whether or not the requester has been granted sufficient authority to take selected actions with respect to that resource object. In a preferred embodiment of the present invention, each access control profile may include access control information relating to a selected user; a selected resource object; a selected group of users; a specified level of authority associated with a selected user; a selected set of resource objects; or, a predetermined set of resource objects and a selected list of users each authorized to access at least a portion of said predetermined set of resource objects.

=> d ti

US PAT NO: 5,263,165 [IMAGE AVAILABLE] L10: 1 of 1
TITLE: System for providing user access control within a distributed data processing system having multiple resource managers

=> d his

(FILE 'USPAT' ENTERED AT 16:00:27 ON 24 AUG 95)
L1 828 S 395/700/CCLS
L2 1426 S 364/280,281.6/CCLS
L3 1972 S L1 OR L2
L4 1281 S ACCESS AND SERVER AND (USER OR CLIENT)
L5 336 S ACCESS(P)SERVER(P)(USER OR CLIENT)

```
L6            39 S RESOURCE(P)L5
L7           669 S RESOURCE(10A)(INFORMATION OR DESCRIPTION OR STUB)
L8           102 S L4 AND L7
L9            23 S PROFILE(5A)RESOURCE
L10            1 S L9/AB

=> d kwic l9 1-25
     23 ANSWERS ARE AVAILABLE. SPECIFIED ANSWER NUMBER EXCEEDS ANSWER SET
SIZE
ENTER ANSWER NUMBER OR RANGE (1):1-23

US PAT NO:     5,428,782 [IMAGE AVAILABLE]            L9: 1 of 23

DETDESC:

DETD(84)

 Many . . . which utilize system-dependent techniques to communicate
with the run-time functions linked to application transactions. These
services include documentation retrieval system,  profile  management,
and distributed  resource  control. The DAA services are implemented as
servers; functions within application transactions are clients.

DETDESC:

DETD(246)

 DRCVDS--VIEW . . . Distributed Resource Control View Distribution
Services, also referred to as the View server. The VIEW server combines
the functions of  profile  management, distribution services, and
 resource  management (including TP monitor-controlled transactions and
terminals) into a single interface for DAA transactions (via IET). View
distribution is implemented. . .

DETDESC:

DETD(247)

 The view server does not have its own data base; it uses the SNADS data
base for  resource  control information and the  profile  management
data base (user and roll files) for view storage and retrieval.

US PAT NO:     5,421,015 [IMAGE AVAILABLE]            L9: 2 of 23

DETDESC:

DETD(343)

 RESPACK . . . be described, becomes associated with that program,
function, or user through a corresponding modification of the program's
or user's associated  profile . Thereafter, the customized  resource
will be used in place of the original, unmodified resource.

DETDESC:

DETD(740)

 Descriptor . . . in, for example, the index of resources in the
resource editor and, for those users or object managers using the
 resource , the user or program  profile .
```

DETDESC:

DETD(742)

User . . . resource. System modifiable flag field 720 is similar but indicates whether the system administrator will be able to customize the resource in the system profile .

DETDESC:

DETD(754)

When . . . in the entry indicates that a customized version of the resource could exist, then RESPACK determines whether the user's User Profile includes a customized resource with the specified Resource ID and the current Customization ID. RESPACK will retrieve the customized resource, if one exists, or. . .

DETDESC:

DETD(756)

When . . . version of the resource is created. This customized copy of the resource is stored as part of that user's User Profile . Each customized copy of a resource has associated with it a Customization ID and a Resource ID. The Resource ID is the same as that of. . .

DETDESC:

DETD(1095)

This function locates a resource in a user profile or resource file and checks its type against the specified type. If the types match, it allocates space from the default heap. . .

DETDESC:

DETD(1097)

In the case of success, the resource is read from the user profile , or one of the open resource files if it is not found in the user profile. When the caller is finished with the resource, it should. . .

US PAT NO: 5,421,012 [IMAGE AVAILABLE] L9: 3 of 23

DETDESC:

DETD(344)

RESPACK . . . be described, becomes associated with that program, function, or user through a corresponding modification of the program's or user's associated profile . Thereafter, the customized resource will be used in place of the original, unmodified resource.

DETDESC:

DETD(738)

Descriptor . . . in, for example, the index of resources in the resource editor and, for those users or object managers using the resource , the user or program profile .

DETDESC:

DETD(740)

User . . . resource. System modifiable flag field 720 is similar but indicates whether the system administrator will be able to customize the resource in the system profile .

DETDESC:

DETD(752)

When . . . in the entry indicates that a customized version of the resource could exist, then RESPACK determines whether the user's User Profile includes a customized resource with the specified Resource ID and the current Customization ID. RESPACK will retrieve the customized resource, if one exists, or. . .

DETDESC:

DETD(754)

When . . . version of the resource is created. This customized copy of the resource is stored as part of that user's User Profile . Each customized copy of a resource has associated with it a Customization ID and a Resource ID. The Resource ID is the same as that of. . .

DETDESC:

DETD(1091)

This function locates a resource in a user profile or resource file and checks its type against the specified type. If the types match, it allocates space from the default heap. . .

DETDESC:

DETD(1093)

In the case of success, the resource is read from the user profile , or one of the open resource files if it is not found in the user profile. When the caller is finished with the resource, it should. . .

US PAT NO: 5,416,694 [IMAGE AVAILABLE] L9: 4 of 23

DETDESC:

DETD(45)

Establishing . . . the skills database. Each data record in the training resources database is preferably comprised of data fields representing a training resource profile of preselected information for each training source. Examples of data held in this database would include training courses offered by. . .

CLAIMS:

CLMS(1)

What . . .
skill in said skills database each data record in said training resources database comprised of data fields representing a training resource profile of preselected information for each training source;
building at the computer host system, a client desired position profile having a plurality. . .

CLAIMS:

CLMS(10)

10. . . .
skill in said skills database each data record in said training resources database comprised of data fields representing a training resource profile of preselected information for each training source;
building at the computer host system, a client desired position profile having a plurality. . .

US PAT NO: 5,375,244 [IMAGE AVAILABLE] L9: 5 of 23

CLAIMS:

CLMS(18)

18. . . .
a probability of ineligibility based upon a degree of resemblance between said attributes of said particular user and said second profile , and
allowing access to the resource by said particular user if the magnitude of said first signal is greater than the magnitude of said second signal.

US PAT NO: 5,369,778 [IMAGE AVAILABLE] L9: 6 of 23

DETDESC:

DETD(344)

RESPACK . . . be described, becomes associated with that program, function, or user through a corresponding modification of the program's or user's associated profile . Thereafter, the customized resource will be used in place of the original, unmodified resource.

DETDESC:

DETD(738)

Descriptor . . . in, for example, the index of resources in the resource editor and, for those users or object managers using the resource , the user or program profile .

DETDESC:

DETD(740)

User . . . resource. System modifiable flag field 720 is similar but indicates whether the system administrator will be able to customize the resource in the system profile .

DETDESC:

DETD(752)

When . . . in the entry indicates that a customized version of the resource could exist, then RESPACK determines whether the user's User Profile includes a customized resource with the specified Resource ID and the current Customization ID. RESPACK will retrieve the customized resource, if one exists, or. . .

DETDESC:

DETD(754)

When . . . version of the resource is created. This customized copy of the resource is stored as part of that user's User Profile . Each customized copy of a resource has associated with it a Customization ID and a Resource ID. The Resource ID is the same as that of. . .

DETDESC:

DETD(1093)

This function locates a resource in a user profile or resource file and checks its type against the specified type. If the types match, it allocates space from the default heap. . .

DETDESC:

DETD(1095)

In the case of success, the resource is read from the user profile , or one of the open resource files if it is not found in the user profile. When the caller is finished with the resource, it should. . .

CLAIMS:

CLMS(1)

What . . .
providing an original version of the resource,
(C) creating a modified version of the resource,
(D) storing the modified version of the resource in a user profile associated with a user and containing information pertaining to the user in the data processing system, and when the resource. . . the data processing system checking the user profile for a version of the required resource,
(G) if a version of such resource exists in the user profile , then the data processing system automatically providing to the program, and without intervention by the user of the program, the version of the resource from the user profile , otherwise the data processing system automatically providing to the program, and without intervention by the user of the program, the. . .

US PAT NO: 5,321,841 [IMAGE AVAILABLE] L9: 7 of 23

DETDESC:

DETD(30)

The . . . and control. A user object 230, which appears at the highest level of the User-Job-Process-Thread (UJPT) hierarchy, defines the security profile and resource quotas/limits for its underlying objects. The user object 230 also stores a pointer to the job object 232 for the. . .

US PAT NO: 5,303,379 [IMAGE AVAILABLE] L9: 8 of 23

DETDESC:

DETD(344)

RESPACK . . . be described, becomes associated with that program, function, or user through a corresponding modification of the program's or user's associated profile . Thereafter, the customized resource will be used in place of the original, unmodified resource.

DETDESC:

DETD(739)

Descriptor . . . in, for example, the index of resources in the resource editor and, for those users or object managers using the resource , the user or program profile .

DETDESC:

DETD(741)

User . . . resource. System modifiable flag field 720 is similar but indicates whether the system administrator will be able to customize the resource in the system profile .

DETDESC:

DETD(753)

When . . . in the entry indicates that a customized version of the resource could exist, then RESPACK determines whether the user's User Profile includes a customized resource with the specified Resource ID and the current Customization ID. RESPACK will retrieve the customized resource, if one exists, or. . .

DETDESC:

DETD(755)

When . . . version of the resource is created. This customized copy of the resource is stored as part of that user's User Profile . Each customized copy of a resource has associated with it a Customization ID and a Resource ID. The Resource ID is the same as that of. . .

DETDESC:

DETD(1092)

This function locates a resource in a user profile or resource file and checks its type against the specified type. If the types match, it allocates space from the default heap. . .

DETDESC:

DETD(1094)

In the case of success, the resource is read from the user profile , or one of the open resource files if it is not found in the user profile. When the caller is finished with the resource, it should. . .

US PAT NO: 5,297,283 [IMAGE AVAILABLE] L9: 9 of 23

DETDESC:

DETD(31)

The . . . and control. A user object 230, which appears at the highest level of the User-Job-Process-Thread (UJPT) hierarchy, defines the security profile and resource quotas/limits for its underlying objects. The user object 230 also stores a pointer to the job object 232 for the. . .

US PAT NO: 5,263,165 [IMAGE AVAILABLE] L9: 10 of 23

ABSTRACT:
The . . . that resource manager. A resource manager may utilize this communication technique to retrieve, modify, or delete a selected access control profile , as desired. Further, the resource manager may utilize this communication technique to control access to a resource object by utilizing the information contained within the. . .

SUMMARY:

BSUM(16)

The . . . that resource manager. A resource manager may utilize this communication technique to retrieve, modify, or delete a selected access control profile , as desired. Further, the resource manager may utilize this communication technique to control access to a resource object by utilizing the information contained within the. . .

DETDESC:

DETD(12)

With . . . is illustrated, the process begins at block 60 and thereafter passes to block 62, which depicts the system administrator or resource manager communicating an Access Profile Command to the Reference Monitor service. By "Access Profile Command" what is meant is a command which will cause an. . .

US PAT NO: 5,263,158 [IMAGE AVAILABLE] L9: 11 of 23

DETDESC:

DETD(16)

Next, . . . Reference Monitor applications which may exist within the distributed data processing system to determine whether or not an access control profile exists for the resource object or user in question. Block 78 then illustrates the logging of this access attempt at the Reference Monitor application.. . .

US PAT NO: 5,263,157 [IMAGE AVAILABLE] L9: 12 of 23

DETDESC:

DETD(13)

In . . . additional resource objects require access control profiles, the process passes to block 68 which illustrates the establishment by an associated resource manager of an access control profile for one or more users within the distributed data processing system. Thereafter, block 70 illustrates the storing of the access. . .

DETDESC:

DETD(14)

Finally, . . . block 84 which illustrates the query of the nearest Reference Monitor application to determine whether or not an access control profile exists for the resource object or user in question.

DETDESC:

DETD(15)

Block . . . This determination is, as those skilled in the art will appreciate, simply a matter of comparing the defined access control profile with the parameters of the resource object and the user in question. Thereafter, as illustrated in block 90, if the determination of block 88 so permits,. . .

CLAIMS:

CLMS(1)

I . . .
resource objects, wherein access to said particular resource object is controlled by said selected resource manager;
transmitting a selected access control profile associated with said particular resource object from said associated reference monitor service to said selected one of said resource managers if said selected access control. . .

CLAIMS:

CLMS(3)

3. . . .
resource objects, wherein access to said particular resource object is controlled by said selected resource manager;
transmitting a selected access control profile associated with said particular resource object from said associated reference monitor

service to said selected one of said resource managers if said selected access control. . .

CLAIMS:

CLMS(5)

5. . . .
wherein access to said particular resource object is controlled by said selected resource manager;
means for transmitting a selected access control profile associated with said particular resource object from said associated reference monitor service to said selected one of said resource managers if said selected access control. . .

CLAIMS:

CLMS(6)

6. . . .
wherein access to said particular resource object is controlled by said selected resource manager;
means for transmitting a selected access control profile associated with said particular resource object from said associated reference monitor service to said selected one of said resource managers if said selected access control. . .

US PAT NO: 5,261,080 [IMAGE AVAILABLE] L9: 13 of 23

DETDESC:

DETD(343)

RESPACK . . . be described, becomes associated with that program, function, or user through a corresponding modification of the program's or user's associated profile . Thereafter, the customized resource will be used in place of the original, unmodified resource.

DETDESC:

DETD(740)

Descriptor . . . in, for example, the index of resources in the resource editor and, for those users or object managers using the resource , the user or program profile .

DETDESC:

DETD(742)

User . . . resource. System modifiable flag field 720 is similar but indicates whether the system administrator will be able to customize the resource in the system profile .

DETDESC:

DETD(754)

When . . . in the entry indicates that a customized version of the resource could exist, then RESPACK determines whether the user's User

Profile includes a customized resource with the specified Resource ID and the current Customization ID. RESPACK will retrieve the customized resource, if one exists, or. . .

DETDESC:

DETD(756)

When . . . version of the resource is created. This customized copy of the resource is stored as part of that user's User Profile . Each customized copy of a resource has associated with it a Customization ID and a Resource ID. The Resource ID is the same as that of. . .

DETDESC:

DETD(1094)

This function locates a resource in a user profile or resource file and checks its type against the specified type. If the types match, it allocates space from the default heap. . .

DETDESC:

DETD(1096)

In the case of success, the resource is read from the user profile , or one of the open resource files if it is not found in the user profile. When the caller is finished with the resource it should. . .

US PAT NO: 5,226,161 [IMAGE AVAILABLE] L9: 14 of 23

DETDESC:

DETD(343)

RESPACK . . . be described, becomes associated with that program, function, or user through a corresponding modification of the program's or user's associated profile . Thereafter, the customized resource will be used in place of the original, unmodified resource.

DETDESC:

DETD(737)

Descriptor . . . in, for example, the index of resources in the resource editor and, for those users or object managers using the resource , the user or program profile .

DETDESC:

DETD(739)

User . . . resource. System modifiable flag field 720 is similar but indicates whether the system administrator will be able to customize the resource in the system profile .

DETDESC:

DETD(751)

When . . . in the entry indicates that a customized version of the resource could exist, then RESPACK determines whether the user's User Profile includes a customized resource with the specified Resource ID and the current Customization ID. RESPACK will retrieve the customized resource, if one exists, or. . .

DETDESC:

DETD(753)

When . . . version of the resource is created. This customized copy of the resource is stored as part of that user's User Profile . Each customized copy of a resource has associated with it a Customization ID and a Resource ID. The Resource ID is the same as that of. . .

DETDESC:

DETD(1093)

This function locates a resource in a user profile or resource file and checks its type against the specified type. If the types match, it allocates space from the default heap. . .

DETDESC:

DETD(1095)

In the case of success, the resource is read from the user profile , or one of the open resource files if it is not found in the user profile. When the caller is finished with the resource, it should. . .

US PAT NO: 5,206,951 [IMAGE AVAILABLE] L9: 15 of 23

DETDESC:

DETD(341)

RESPACK . . . be described, becomes associated with that program, function, or user through a corresponding modification of the program's or user's associated profile . Thereafter, the customized resource will be used in place of the original, unmodified resource.

DETDESC:

DETD(736)

Descriptor . . . in, for example, the index of resources in the resource editor and, for those users or object managers using the resource , the user or program profile .

DETDESC:

DETD(738)

User . . . resource. System modifiable flag field 720 is similar but indicates whether the system administrator will be able to customize the resource in the system profile .

DETDESC:

DETD(750)

When . . . in the entry indicates that a customized version of the resource could exist, then RESPACK determines whether the user's User Profile includes a customized resource with the specified Resource ID and the current Customization ID. RESPACK will retrieve the customized resource, if one exists, or. . .

DETDESC:

DETD(752)

When . . . version of the resource is created. This customized copy of the resource is stored as part of that user's User Profile . Each customized copy of a resource has associated with it a Customization ID and a Resource ID. The Resource ID is the same as that of. . .

DETDESC:

DETD(1092)

This function locates a resource in a user profile or resource file and checks its type against the specified type. If the types match, it allocates space from the default heap. . .

DETDESC:

DETD(1094)

In the case of success, the resource is read from the user profile , or one of the open resource files if it is not found in the user profile. When the caller is finished with the resource, it should. . .

US PAT NO: 5,187,790 [IMAGE AVAILABLE] L9: 16 of 23

DETDESC:

DETD(31)

The . . . and control. A user object 230, which appears at the highest level of the User-Job-Process-Thread (UJPT) hierarchy, defines the security profile and resource quotas/limits for its underlying objects. The user object 230 also stores a pointer to the job object 232 for the. . .

US PAT NO: 5,136,712 [IMAGE AVAILABLE] L9: 17 of 23

DETDESC:

DETD(31)

The . . . and control. A user object 230, which appears at the highest level of the User-Job-Process-Thread (UJPT) hierarchy, defines the security profile and resource quotas/limits for its underlying objects. The user object 230 also stores a pointer to the job object 232 for the. . .

US PAT NO: 5,129,084 [IMAGE AVAILABLE] L9: 18 of 23

DETDESC:

DETD(30)

The . . . and control. A user object 230, which appears at the highest level of the User-Job-Process-Thread (UJPT) hierarchy, defines the security profile and resource quotas/limits for its underlying objects. The user object 230 also stores a pointer to the job object 232 for the. . .

US PAT NO: 5,129,083 [IMAGE AVAILABLE] L9: 19 of 23

DETDESC:

DETD(31)

The . . . and control. A user object 230, which appears at the highest level of the User-Job-Process-Thread (UJPT) hierarchy, defines the security profile and resource quotas/limits for its underlying objects. The user object 230 also stores a pointer to the job object 232 for the. . .

US PAT NO: 5,057,996 [IMAGE AVAILABLE] L9: 20 of 23

DETDESC:

DETD(31)

The . . . and control. A user object 230, which appears at the highest level of the User-Job-Process-Thread (UJPT) hierarchy, defines the security profile and resource quotas/limits for its underlying objects. The user object 230 also stores a pointer to the job object 232 for the. . .

US PAT NO: 5,025,395 [IMAGE AVAILABLE] L9: 21 of 23

CLAIMS:

CLMS(11)

11. . . .
said commands;
pattern data file means for storing a plurality of graphic patterns;
profile data file means for storing a plurality of profile data files corresponding to resource identifiers, each of said profile data files including a plurality of personal record areas, each of which stores user identifying information, an indicator to specify. . .

CLAIMS:

CLMS(12)

12. . . . data processing, comprising:
file means for storing a plurality of profile files prepared for said resources, respectively, each of said profile files including a resource identifier for identifying the resource and at least one graphic pattern representing the resource corresponding to a user code

assigned. . . data processing means including means responsive to said designating means for performing data processing on a resource corresponding to a resource identifier in one of said profile files which includes said graphic pattern specified by said designating means.

US PAT NO: 4,890,227 [IMAGE AVAILABLE] L9: 22 of 23

DETDESC:

DETD(10)

FIG. . . . axis of resources R.sub.1 -R.sub.n and axis of a resource utilization (for example, resource utilization factor per unit time). The resource utilization profile 330, 340 and 350 for the resources at times T.sub.0, T.sub.1 and T.sub.2 are shown.

US PAT NO: 4,768,150 [IMAGE AVAILABLE] L9: 23 of 23

DETDESC:

DETD(5)

Referring . . . subroutine 21 to the OPEN command 31. The "snaopen" subroutine 21 includes a resource parameter 41 that specifies a connection profile name of the resource to be opened. The OPEN command 31 includes a "path" parameter 42 and an "oflag" parameter 43. The "path" parameter. . . also specifies the resource to be opened by specifying the SNA device driver name to be used to open the resource , and by specifying the connection profile name of the resource to be opened. If the "snaopen" subroutine completes successfully, it returns an integer that specifies the connection ID (cid) for. . .

=> t

```
=> s 395/clas
L1        24862 395/CLAS

=> s 364/280,281.6/ccls
          1357 364/280/CCLS
           165 364/281.6/CCLS
L2        1426 364/280,281.6/CCLS
                ((364/280 OR 364/281.6)/CCLS)

=> s 5267235/pn or 5341477/pn or 5036459/pn
             1 5267235/PN
             1 5341477/PN
             1 5036459/PN
L3           3 5267235/PN OR 5341477/PN OR 5036459/PN

=> d ti 1-3

US PAT NO:       5,341,477   [IMAGE AVAILABLE]        L3: 1 of 3
TITLE:          Broker for computer network server selection

US PAT NO:       5,267,235   [IMAGE AVAILABLE]        L3: 2 of 3
TITLE:          Method and apparatus for resource arbitration

US PAT NO:       5,036,459   [IMAGE AVAILABLE]        L3: 3 of 3
TITLE:          Multi-processor computer system with distributed memory
                  and an interprocessor communication mechanism, and
                  method for operating such mechanism

=> d ti,ab 1-3

US PAT NO:       5,341,477   [IMAGE AVAILABLE]        L3: 1 of 3
TITLE:          Broker for computer network server selection

ABSTRACT:
In a computer network, a broker mechanism allocates a plurality of
servers, each having an available resource capacity, to a plurality of
clients for delivering one of several services to the clients. The broker
operates by monitoring a subset of all available servers capable of
delivering the requested service. The allocation is based on developing a
network policy for the plurality of servers by collecting a local policy
for each of the servers. The broker receives client requests for the
services and based on the network policy and available resource capacity
suggests one of the servers, monitors in its subset for that particular
service, to one of the clients making a request. The server suggested
enforces its local policy by not allowing any connections exceeding its
available resource capacity.

US PAT NO:       5,267,235   [IMAGE AVAILABLE]        L3: 2 of 3
TITLE:          Method and apparatus for resource arbitration

ABSTRACT:
The present invention provides a rapid one-to-one match between
requesters that must arbitrate for service from one of a number of
servers. Each requester presents a set of requests, and the requesters
are indifferent to which server is chosen, no priority existing among the
requests seen by a particular server. Requests are presented
synchronously to all servers to which access is desired. Each server
selects precisely one such request, preferably randomly, and asserts a
response signal so stating to all requesters. Each requester then selects
precisely one incoming grant responses (if any there are), and de-asserts
```

requests to all other servers. This iteration is repeated for a predetermined number of cycles, at which time substantially most of the requested matches will have been made. The iteration algorithm is preferably implemented with choice units, multiplexers, registers and logic units, all of which may be obtained commercially.

US PAT NO: 5,036,459 [IMAGE AVAILABLE] L3: 3 of 3
TITLE: Multi-processor computer system with distributed memory and an interprocessor communication mechanism, and method for operating such mechanism

ABSTRACT:
A method and process for operating an interprocess communication mechanism in a multi processor computer system are described. If a sender node needs sending a message to a receiver node, it accesses the latter for available storage space. If available, the message is transferred and the sender node may resume processing. If nonavailable the transferring is deferred. In either case the message is queued to any message either awaiting processing at the receiver node or awaiting transfer, in that at the instant when such transfer was necessary, no storage space had been available. If the receiver node wants to process a message, it accesses the least recent item of the message queue and unlinks it, while belated transferring is now executed, if necessary.
Thereupon the original sender node was still kept waiting, it is now allowed to resume processing. Generally only two communication operations are required per message. Only in case of inavailability of storage space, the message in question needs four communication operations.

=> SET HIGHLIGHTING OFF
SET COMMAND COMPLETED

=> s access list#/ab
25990 ACCESS/AB
1502 LIST#/AB
L4 11 ACCESS LIST#/AB
((ACCESS(W)LIST#)/AB)

=> d ti 1-11

US PAT NO: 5,426,748 [IMAGE AVAILABLE] L4: 1 of 11
TITLE: Guest/host extended addressing method and means with contiguous access list entries

US PAT NO: 5,414,852 [IMAGE AVAILABLE] L4: 2 of 11
TITLE: Method for protecting data in a computer system

US PAT NO: 5,390,312 [IMAGE AVAILABLE] L4: 3 of 11
TITLE: Access look-aside facility

US PAT NO: 5,381,537 [IMAGE AVAILABLE] L4: 4 of 11
TITLE: Large logical addressing method and means

US PAT NO: 5,361,356 [IMAGE AVAILABLE] L4: 5 of 11
TITLE: Storage isolation with subspace-group facility

US PAT NO: 5,276,901 [IMAGE AVAILABLE] L4: 6 of 11
TITLE: System for controlling group access to objects using group access control folder and group identification as individual user

```
US PAT NO:      5,271,007 [IMAGE AVAILABLE]              L4: 7 of 11
TITLE:          Network system having controlled access to available
                  resources

US PAT NO:      5,151,994 [IMAGE AVAILABLE]              L4: 8 of 11
TITLE:          Distributed fair arbitration system using separate grant
                  and request lines for providing access to data
                  communication bus

US PAT NO:      5,023,773 [IMAGE AVAILABLE]              L4: 9 of 11
TITLE:          Authorization for selective program access to data in
                  multiple address spaces

US PAT NO:      4,979,098 [IMAGE AVAILABLE]              L4: 10 of 11
TITLE:          Multiple address space token designation, protection
                  controls, designation translation and lookaside

US PAT NO:      4,945,480 [IMAGE AVAILABLE]              L4: 11 of 11
TITLE:          Data domain switching on program address space switching
                  and return

=> d his

     (FILE 'USPAT' ENTERED AT 13:03:43 ON 24 AUG 95)
L1         24862 S 395/CLAS
L2          1426 S 364/280,281.6/CCLS
L3             3 S 5267235/PN OR 5341477/PN OR 5036459/PN
                 SET HIGHLIGHTING OFF
L4            11 S ACCESS LIST#/AB

=> s access/ab
L5         25990 ACCESS/AB

=> s l1 or l2
L6         24927 L1 OR L2

=> s l5 and l6
L7          3254 L5 AND L6

=> s access/ti
L8          3490 ACCESS/TI

=> s l6 and l8
L9           728 L6 AND L8

=> s (object# and access and server and (user or client))/ab
           33281 OBJECT#/AB
           25990 ACCESS/AB
             349 SERVER/AB
           25165 USER/AB
             154 CLIENT/AB
L10            2 (OBJECT# AND ACCESS AND SERVER AND (USER OR CLIENT))/AB

=> d ti 1-2

US PAT NO:      5,321,841 [IMAGE AVAILABLE]              L10: 1 of 2
TITLE:          System for determining the rights of object access for a
                  server process by combining them with the rights of the
                  client process
```

US PAT NO: 5,187,790 [IMAGE AVAILABLE] L10: 2 of 2
TITLE: Server impersonation of client processes in an object based computer operating system

=> d ti,ab 1-2

US PAT NO: 5,321,841 [IMAGE AVAILABLE] L10: 1 of 2
TITLE: System for determining the rights of object access for a server process by combining them with the rights of the client process

ABSTRACT:
In a multitasking, multiuser computer system, a server process temporarily impersonates the characteristics of a client process when the client process preforms a remote procedure call on the server process. Each process has an identifier list with a plurality of identifiers that characterize the process. The server process generates a new identifier list which is either the same as the client process's list, or is the union of the server's and the client's lists. Each object in the system can have an access control list which defines the identifiers that a process must have in order to access the object. The operation system has access checking software for enabling a selected process access to a specified object when the identifiers for the process match the list of identifiers in the access control list of the specified object. The server can therefore access all objects accessible to the client while the server is working for the client. The server can restore its original identifier list after completing the services that it performs for the client.

US PAT NO: 5,187,790 [IMAGE AVAILABLE] L10: 2 of 2
TITLE: Server impersonation of client processes in an object based computer operating system

Same

ABSTRACT:
In a multitasking, multiuser computer system, a server process temporarily impersonates the characteristics of a client process when the client process preforms a remote procedure call on the server process. Each process has an identifier list with a plurality of identifiers that characterize the process. The server process generates a new identifier list which is either the same as the client process's list, or is the union of the server's and the client's lists. Each object in the system can have an access control list which defines the identifiers that a process must have in order to access the object. The operation system has access checking software for enabling a selected process access to a specified object when the identifers for the process match the list of identifiers in the access control list of the specified object. The server can therefore access all objects accessible to the client while the server is working for the client. The server can restore its original identifier list after completing the services that it performs for the client.

=> ç

=> d ti,ab 19 10

US PAT NO: 5,263,165 [IMAGE AVAILABLE] L9: 10 of 23
TITLE: System for providing user access control within a distributed data processing system having multiple resource managers

ABSTRACT:
The method of the present invention may be utilized to provide user access control for a plurality of resource objects within a distributed data processing system having a plurality of resource managers. A reference monitor service is established and a plurality of access control profiles are stored therein. Thereafter, selected access control profile information may be communicated between the reference monitor service and a resource manager in response to an attempted access of a particular resource object controlled by that resource manager. A resource manager may utilize this communication technique to retrieve, modify, or delete a selected access control profile , as desired. Further, the resource manager may utilize this communication technique to control access to a resource object by utilizing the information contained within the access control profile to determine if the requester is authorized to access the resource object and whether or not the requester has been granted sufficient authority to take selected actions with respect to that resource object. In a preferred embodiment of the present invention, each access control profile may include access control information relating to a selected user; a selected resource object; a selected group of users; a specified level of authority associated with a selected user; a selected set of resource objects; or, a predetermined set of resource objects and a selected list of users each authorized to access at least a portion of said predetermined set of resource objects.

=> S

=> d ti,ab l9 12

US PAT NO: 5,263,157 [IMAGE AVAILABLE] L9: 12 of 23
TITLE: Method and system for providing user access control within a distributed data processing system by the exchange of access control profiles

ABSTRACT:
A method is disclosed for providing user access control for a plurality of resource objects within a distributed data processing system having a plurality of resource managers. A reference monitor service is established and a plurality of access control profiles are stored therein. Thereafter, selected access control profiles are exchanged between the reference monitor service and a resource manager in response to an attempted access of a particular resource object controlled by that resource manager. The resource manager may then control access to the resource object by utilizing the exchanged access control profile. In a preferred embodiment of the present invention, each access control profile may include access control information relating to a selected user; a selected resource object; a selected group of users; a selected set of resource objects; or, a predetermined set of resource objects and a selected group of users.

=>

=> d ti,in,ab l9 7

US PAT NO: 5,321,841 [IMAGE AVAILABLE] L9: 7 of 23
TITLE: System for determining the rights of object access for a server process by combining them with the rights of the client process
INVENTOR: Jeffrey A. East, Aptos, CA
James J. Walker, Redmond, WA
Steven M. Jenness, Redmond, WA
Mark C. Ozur, Redmond, WA
James W. Kelly, Jr., Redmond, WA

ABSTRACT:
In a multitasking, multiuser computer system, a server process temporarily impersonates the characteristics of a client process when the client process preforms a remote procedure call on the server process. Each process has an identifier list with a plurality of identifiers that characterize the process. The server process generates a new identifier list which is either the same as the client process's list, or is the union of the server's and the client's lists. Each object in the system can have an access control list which defines the identifiers that a process must have in order to access the object. The operation system has access checking software for enabling a selected process access to a specified object when the identifiers for the process match the list of identifiers in the access control list of the specified object. The server can therefore access all objects accessible to the client while the server is working for the client. The server can restore its original identifier list after completing the services that it performs for the client.

=>

65926/8 abandoned, cont. to 8/432372
Chernick, et al.
title: "Communications On A Network"
old Title: "Object Procedure Messaging Facility"
title should be: "Method For Selecting Server Object To Service Client Object Requests On A Network"
Filing date: 1993 May 21
pct: pct/us 94/05876
summary: invention is in an object-oriented system where clients and servers have stubs - little programs which interface a given client with a given server. message requests by clients are queued and when a server is free, its stub notifies the queue and if a message is waiting for its server, it gets it and the server responds to the request.
status: 1 (94.6.14), 2F (94.10.25)
note: next time use
Thacker; U.S. pat. 5,267,235; "Method And Apparatus For Resource Arbitration"
Pitkin, et al.; U.S. pat. 5,341,477; "Broker For Computer Network Server Selection"
used:
den Haan, et al.; U.S. pat. 5,036,459; "multi-processor computer system with distributed memory and an interprocessor communication mechanism, and method for operating such mechanism"
cited:
Bednar, Jr., et al.; U.S. pat. 4,630,196; "store and forward facility for use in multiprocessing environment"
Rupp; U.S. pat. 5,321,808; "dual process display server"
used(2):
Johnson, et al.; U.S. pat. 5,133,053; "Interprocess Communication Queue Location Transparency"
cited(2):
Andrade, et al.; U.S. pat. 5,265,250; "Apparatus And Methods For Performing An Application-Defined Operation On Data As Part Of A System-Defined Operation On The Data"
Bednar, Jr., et al.; U.S. pat. 4,630,196; "Store And Forward Facility For Use In Multiprocessing Environment"
Dally, et al.; U.S. pat. 5,212,778; "Message-Driven Processor In A Concurrent Computer"
Foss, et al.; U.S. pat. 5,335,347; "Method And Apparatus For Scoped Interprocess Message Switching"
Gerety, et al.; U.S. pat. 5,212,792; "Method And Apparatus For Controlling Execution Of Tools In A Computer-Aided Software Engineering System"
Priven, et al.; U.S. pat. 5,327,559; "Remote And Batch Processing In An Object Oriented Programming System"
Row, et al.; U.S. pat. 5,355,453; "Parallel I/O Network File Server Architecture"
Simor; U.S. pat. 5,165,018; "Self-Configuration Of Nodes In A Distributed Message-Based Operating System"

=> d ti,in,ab l9 3

US PAT NO: 5,421,012 [IMAGE AVAILABLE] L9: 3 of 23
TITLE: Multitasking computer system for integrating the operation of different application programs which manipulate data objects of different types
INVENTOR: Dana Khoyi, Dracut, MA
Marc S. Soucie, Tyngsboro, MA
Carolyn E. Surppenant, Dracut, MA
Laura O. Stern, Woburn, MA
Ly-Huong T. Pham, Chelmsford, MA

ABSTRACT:
An object based data processing system including an extensible set of object types and a corresponding set of "object managers" wherein each object manager is a program for operating with the data stored in a corresponding type of object. The object managers in general support at least a standard set of operations. Any program can effect performance of these standard operations on objects of any type by making an "invocation" request. In response to an invocation request, object management services (which are available to all object managers) identifies and invokes an object manager that is suitable for performing the requested operation on the specified type of data. A mechanism is provided for linking data from one object into another object. A object catalog includes both information about objects and about links between objects. Data interchange services are provided for communicating data between objects of different types, using a set of standard data interchange formats. A matchmaker facility permits two processes that are to cooperate in a data interchange operation identify each other and to identify data formats they have in common. A facility is provided for managing shared data "resources". Customized versions of resources can be created and co-exist with standard resources. A resource retrieval function determines whether a customized or a standard resource is to be returned in response to each request for a resource.

=> 3
=> d ti,in,ab l9 2

US PAT NO: 5,421,015 [IMAGE AVAILABLE] L9: 2 of 23
TITLE: Multitasking system having an application integration mechanism for linking differently typed data objects
INVENTOR: Dana Khoyi, Dracut, MA
Marc San Soucie, Tyngsboro, MA
Carolyn E. Surprenant, Dracut, MA
Laura O. Stern, Woburn, MA
Ly-Huong T. Pham, Chelmsford, MA

ABSTRACT:
An object based data processing system including an extensible set of object types and a corresponding set of "object managers" wherein each object manager is a program for operating with the data stored in a corresponding type of object. The object managers in general support at least a standard set of operations. Any program can effect performance of these standard operations on objects of any type by making an "invocation" request. In response to an invocation request, object management services (which are available to all object managers) identifies and invokes an object manager that is suitable for performing the requested operation on the specified type of data. A mechanism is provided for linking data from one object into another object. A object catalog includes both information about objects and about links between

objects. Data interchange services are provided for communicating data between objects of different types, using a set of standard data interchange formats. A matchmaker facility permits two processes that are to cooperate in a data interchange operation identify each other and to identify data formats they have in common. A facility is provided, for managing shared data "resources". Customized versions of resources can be created and co-exist with standard resources. A resource retrieval function determines whether a customized or a standard resource is to be returned in response to each request for a resource.

=> b

=> d ti,in,ab l9 8

US PAT NO: 5,303,379 [IMAGE AVAILABLE] L9: 8 of 23
TITLE: Link mechanism for linking data between objects and for performing operations on the linked data in an object based system
INVENTOR: Dana Khoyi, Dracut, MA
Marc S. Soucie, Tyngsboro, MA
Carolyn E. Surprenant, Dracut, MA
Laura O. Stern, Woburn, MA
Ly-Huong T. Pham, Chelmsford, MA

ABSTRACT:
An object based data processing system including an extensible set of object types and a corresponding set of "object managers" wherein each object manager is a program for operating with the data stored in a corresponding type of object. The object managers in general support at least a standard set of operations. Any program can effect performance of these standard operations on objects of any type by making an "invocation" request. In response to an invocation request, object management services (which are available to all object managers) identifies and invokes an object manager that is suitable for performing the requested operation on the specified type of data. A mechanism is provided for linking data from one object into another object. A object catalog includes both information about objects and about links between objects. Data interchange services are provided for communicating data between objects of different types, using a set of standard data interchange formats. A matchmaker facility permits two processes that are to cooperate in a data interchange operation identify each other and to identify data formats they have in common. A facility is provided for managing shared data "resources". Customized versions of resources can be created and co-exist with standard resources. A resource retrieval function determines whether a customized or a standard resource is to be returned in response to each request for a resource.

=> J

=> d ti,in,ab l9 6

US PAT NO: 5,369,778 [IMAGE AVAILABLE] L9: 6 of 23
TITLE: Data processor that customizes program behavior by using a resource retrieval capability
INVENTOR: Marc San Soucie, Tyngsboro, MA
Carolyn E. Surprenant, Dracut, MA
Thomas Fitzgerald, Lowell, MA
Susan Walker, Arlington, MA

ABSTRACT:
A data processing system based on an extensible set of typed data objects and a corresponding set of "object managers," each of which is a program for operating with the data stored in a corresponding type of object. The object managers in general support at least a standard set of operations. Any program can effect performance of these standard operations on objects of any type by making a particular request; in response to such a request, an object manager that is suitable for performing the requested operation on the specified type of data is identified and caused to perform the requested operation. A mechanism is provided for linking data from one object into another object. A catalog includes both information about objects and about links between objects. Data interchange services are provided for communicating data between objects of different types, using a set of standard data interchange formats. A facility is provided to permit two processes that are to cooperate in a data interchange operation to identify each other and to identify data formats they have in common. A facility is provided for managing shared data in units of data known as "resources". Customized versions of resources can be created and co-exist with standard versions of the resources. A resource retrieval function determines whether a customized or a standard resource is to be returned in response to each request for a resource.

=> ⊥